FALL SEMESTER (2022-23)

CALCULUS LAB FOR ENGINEERING

BMAT101P

Name : Jayampu Akash Siva Teja

Reg.no : 22BCE1778

SUBMITTED TO:

Dr.A.Berin Greeni

INDEX:

LAB-1 INTRODUCTION TO MATLAB

MATRIX ADDITION-

INPUT:

```
clc
clear all
a=[7 5;2 0]
b=[1 3;9 5]
c=a+b
```

OUTPUT:

```
a =

     7     5
     2     0


b =

     1     3
     9     5


c =

     8    8
     11   5
```

DIFFERENTIATION-

INPUT:

```
clc
clear all
syms x
y=sin(x);
dy=diff(y,x)
```

OUTPUT:

```
dy =

cos(x)
```

INTEGRATION:

INPUT:

```
clc
clear all
syms x
dy=sin(x)
y=int(dy,x)
```

OUTPUT:

```
dy =

sin(x)


y =

-cos(x)
```

LAB-2 PLOTTING A GRAPH IN MATLAB

CIRCLE:

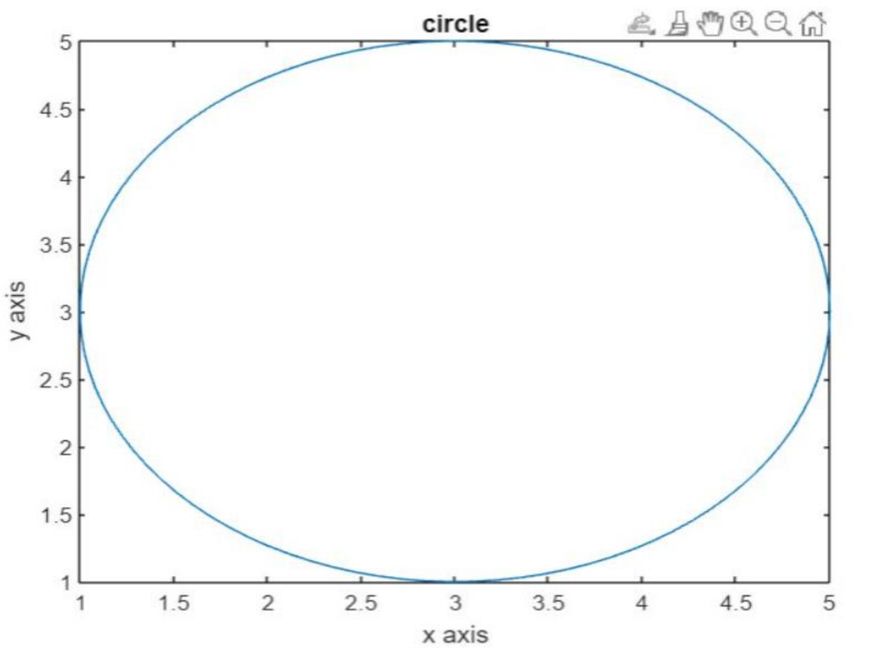
INPUT:

```
clc
clear all
r=input('enter the radius');
a=input('enter x coordinate of circle');
b=input('enter y coordinate of circle');
t=linspace(0,2*pi,100);
x=a+r*cos(t);
y=a+r*sin(t);
plot(x,y)
xlabel('x axis')
ylabel('y axis')
title('circle')
```

OUTPUT:

```
enter the radius
2
enter x coordinate of circle
3
enter y coordinate of circle
4
```

DIAGRAM:

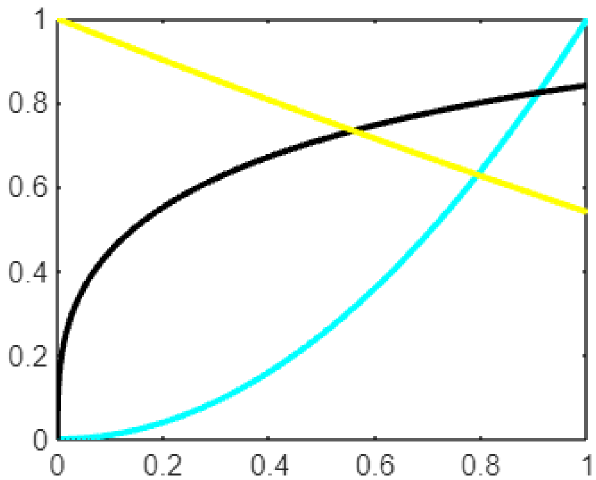


HOLAND,LEGEND:

INPUT:

```
clc
clear all
x=linspace(0,1,100);
plot(x,x.^2,'c','linewidth',2.0);
hold on
plot(x.^3,sin(x),'k','linewidth',2.0);
hold on
plot(x.^2,cos(x),'y','linewidth',2.0);
```

OUTPUT:



SUBPLOTTING:

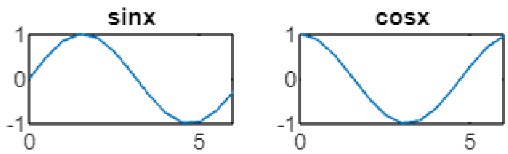
INPUT:

```
clc
clear all
x=0:0.5:2*pi
subplot(3,2,1)
y=sin(x);
plot(x,y)
title('sinx')
subplot(3,2,2)
y=cos(x)
plot(x,y);
title('cosx')
```

OUTPUT:

```
x =

         0    0.5000    1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000    4.5000    5.0000    5.5000    6.0000

y =

    1.0000    0.8776    0.5403    0.0707   -0.4161   -0.8011   -0.9900   -0.9365   -0.6536   -0.2108    0.2837    0.7087    0.9602
```

DIAGRAM:



LAB-3 SOLVING THE FUNCTION IN MATLAB

SOLVING A FUNCTION:

INPUT:

```
syms x
f=input('enter the function')
solve(f)
```

OUTPUT:

```
enter the function
x^2-4

f =

x^2 - 4


ans =

-2
 2
```

FINDING DERIVATIVE AND SOLVING:

INPUT:

```
syms x
f=input('enter the function')
df=diff(f,x)
c1=solve(df)
c=double(c1)
```

OUTPUT :

```
enter the function
cos(x)

f =

cos(x)


df =
```

```
-sin(x)


c1 =

0


c =

     0
```

FINDING DOUBLE DERIVATIVE:

INPUT:

```
syms x real
f=input('enter the function')
df=diff(f,x)
ddf=diff(df,x)
```

OUTPUT:

```
enter the function
x^2-6

f =

x^2 - 6


df =

2*x


ddf =

2
```

MAXIMA AND MINIMA:

INPUT:

```
syms x real
f=input('enter the function')
df=diff(f,x)
ddf=diff(df,x)
c1=solve(df,x)
c=double(c1)
for i = 1 : length(c)
    k=subs(ddf,x,c(i));
    if(k==0)
        sprintf('no conclusion for the point,x=%d',c(i))
    else if (k>0)
            sprintf('min point for the point,x=%d',c(i))
    else
        sprintf('max point')
    end
    end
end
```

OUTPUT:

```
enter the function
cos(x)

f =

cos(x)


df =

-sin(x)


ddf =

-cos(x)


c1 =

0


c =

     0


ans =

    'max point'
```

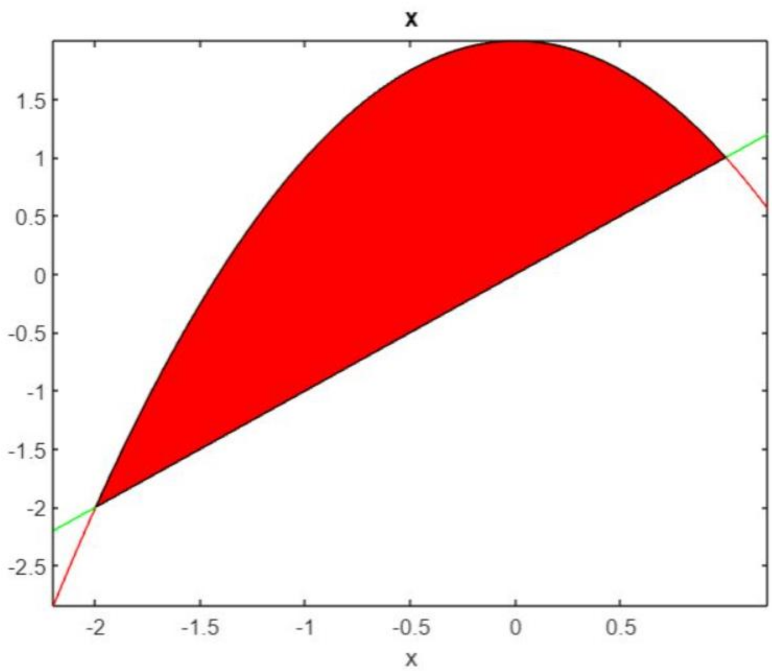
AREA:

INPUT:

```
clear all
syms x
y1=input('enter the upper curve as a function of x:');
y2=input('enter the lower curve as a function of x:');
t=solve(y1-y2);
t=double(t);
A=int(y1-y2,t(1),t(2))
D=[t(1)-0.2 t(2)+0.2];
ez1=ezplot(y1,D);
set(ez1,'color','r')
hold on
ez2=ezplot(y2,D)
set(ez2,'color','g')
xv = linspace(t(1),t(2));
y1v = subs(y1,x,xv);
y2v = subs(y2,x,xv);
x=[xv,xv]
y=[y1v,y2v]
fill(x,y,'r')
```

OUTPUT:

```
enter the upper curve as a function of x:
2-x^2
enter the lower curve as a function of x:
x

A =

9/2
```



LAB-4 PLOTTING GRAPHS OF A FUNCTION IN
TWO VARIABLES

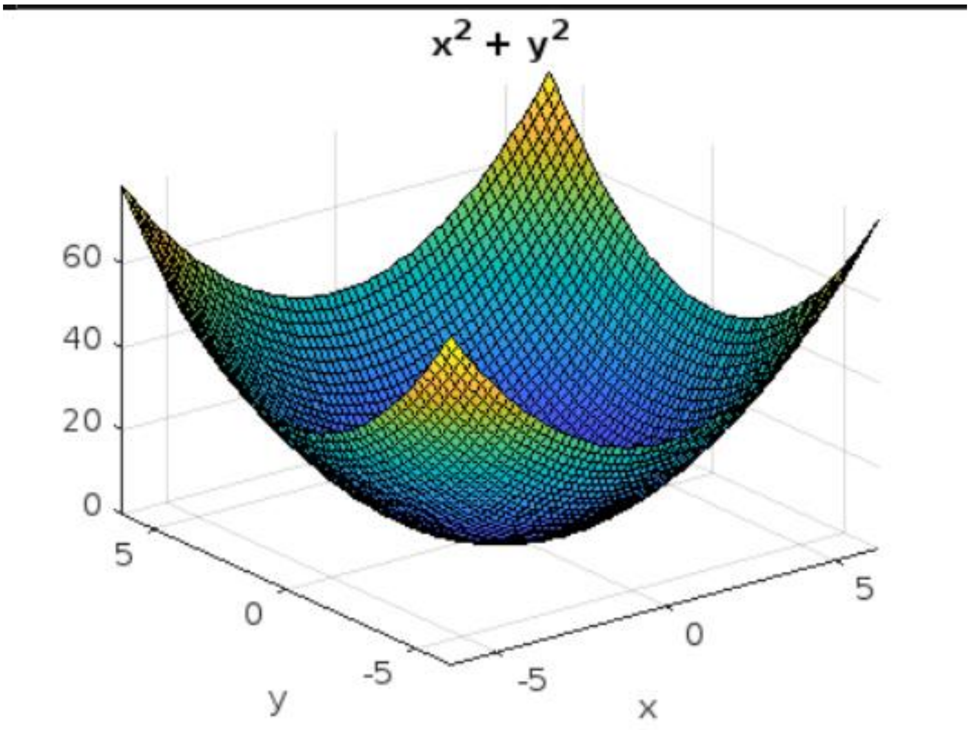
PLOTTING A GARPH OF A FUNCTION IN TWO VARIABLES:

INPUT:

```
clc
clear all
syms x y
f=input('enter the function')
ezsurf(f)
```

OUTPUT:

```
enter the function
x^2+y^2

f =

x^2 + y^2
```

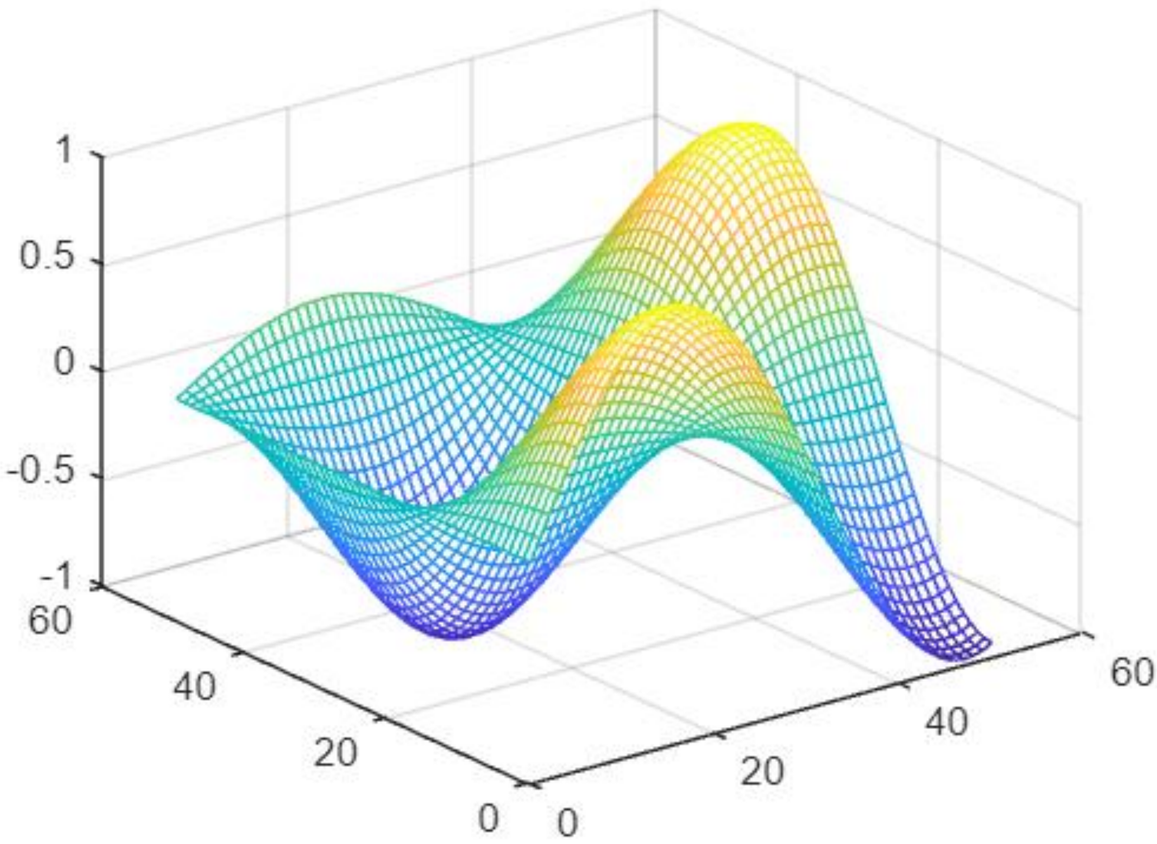


MESHGRID( )

INPUT:

```
clc
clear all
syms x y
[x,y]=meshgrid(0:0.1:5);
z=sin(x).*cos(y);
mesh(z)
```

OUTPUT:



LAGRANGES THEOREM:

INPUT:

```
clc
clear all
syms x y lam real
f=input('Enter the function to be maximised in terms of x and y:');
g=input('Enter the constraint in terms of x and y:');
R=f+lam*g
[alam,ax,ay]=solve(jacobian(R,[x y lam]))
T=subs(f,{x,y},{ax,ay})
for i=1:1:size(T)
    sprintf('The point(x,y) is(%d,%d)',double(ax(i)),double(ay(i)))
    sprintf('The value of the function is %d',double(T(i)))
end
K=sort(T)
F_min=T(1)
F_max=T(length(T))
```

OUTPUT:

```
Enter the function to be maximised in terms of x and y:
2*x+2*x*y+y
```

```
Enter the constraint in terms of x and y:
2*x+y==100

R =

2*x + y + 2*x*y + lam*(2*x + y) == 100*lam + 2*x + y + 2*x*y


alam =

0


ax =

50


ay =

0


T =

100


ans =

    'The value of the function is 100'


K =

100


F_min =

100


F_max =

100
```

MAXIMA AND MINIMA:

INPUT:

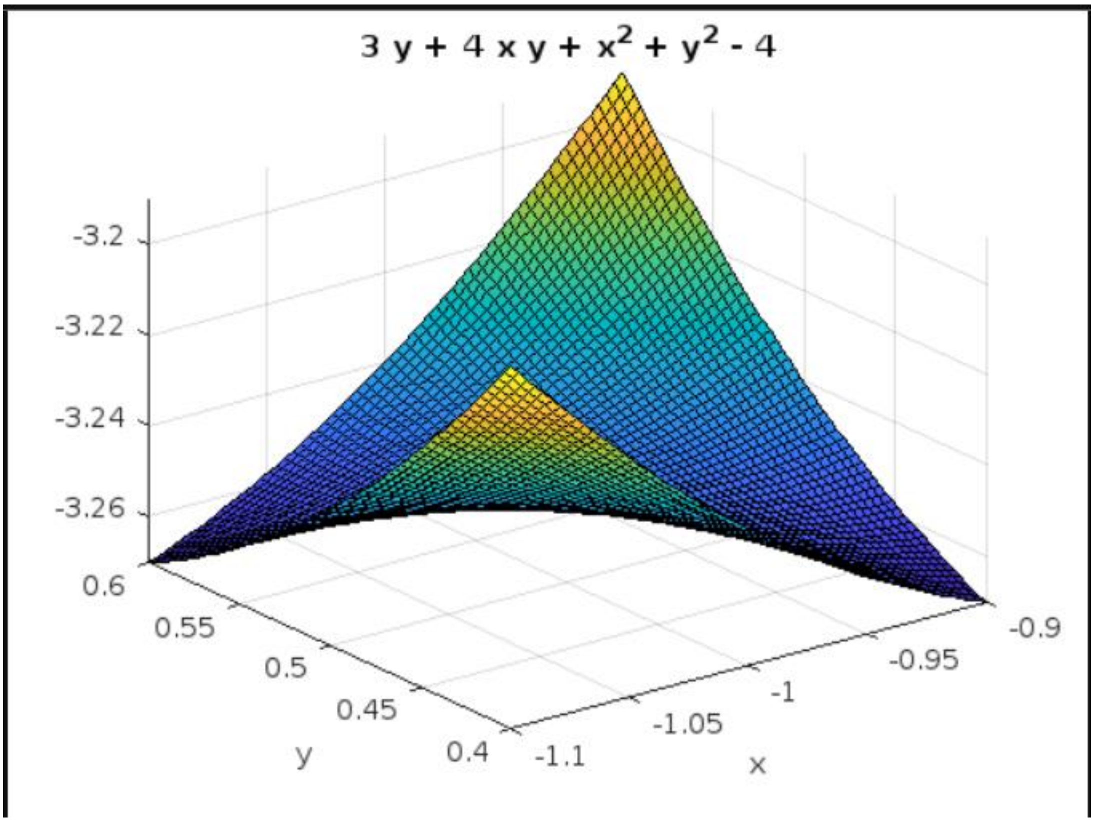
```
clc
clear all
syms x y real
f=input('Enter the function f(x,y)')
fx=diff(f,x);
fy=diff(f,y);
```

```
[ax,ay]=solve(fx,fy);
ax=double(ax);
ay=double(ay);
fxx=diff(fx,x);
fxy=diff(fx,y);
fyy=diff(fy,y);
D=fxx*fyy-fxy^2;
for i=1;size(ax)
    figure
    T1=subs(subs(D,x,ax(i)),y,ay(i));
    T1=double(T1);
    T2=subs(subs(fxx,x,ax(i)),y,ay(i));
    T2=double(T2);
    T3=subs(subs(f,x,ax(i)),y,ay(i));
    T3=double(T3);
    if(T1==0)
        fprintf('The critical point(%d,%d)needs to be investigated
further\n',ax(i))

        else if(T1<0)
                fprintf('The critical point(%d,%d)is a saddle',ax(i),ay(i))

            else if(T2<0)
                fprintf('The maximum point is(%d,%d)',ax(i),ay(i))
                fprintf('The maximum value of function is%d',T3)
            end
        end
    end
end
R=[ax(i)-0.1,ax(i)+0.1,ay(i)-0.1,ay(i)+0.1];
ezsurf(f,R);
hold on
plot3(ax(i),ay(i),T3,'k','markersize',10)
```

OUTPUT:

```
Enter the function f(x,y)
x^2+y^2+4*x*y+3*y-4

f =

x^2 + 4*x*y + y^2 + 3*y - 4


ans =

     1     1

The critical point(-1,5.000000e-01)is a saddle
```

DIAGRAM:



LAB-5:

Syntax-viewSolid

CODE-1:

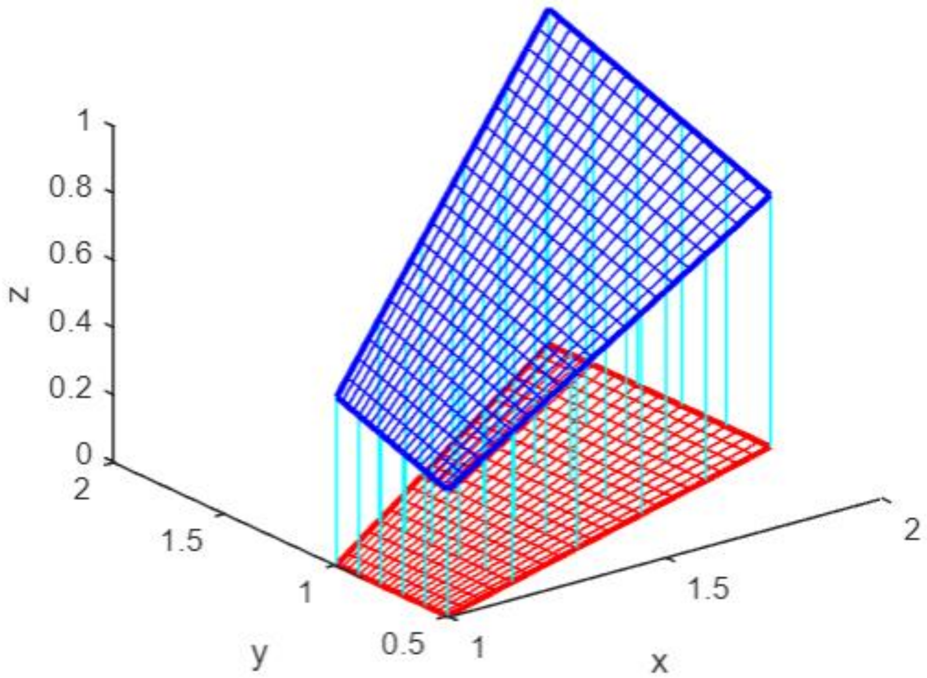
INPUT:

```
syms x y z
f = (x+y)/4
int(int(f,y,x/2,x),x,1,2)
viewSolid(z,0+0*x+0*y,(x+y)/4,y,x/2,x,x,1,2)
```

OUTPUT:

```
f =

x/4 + y/4


ans =

49/96
```

DIAGRAM:



VOLUME OF THE REGION:

Syntax-viewSolidone:

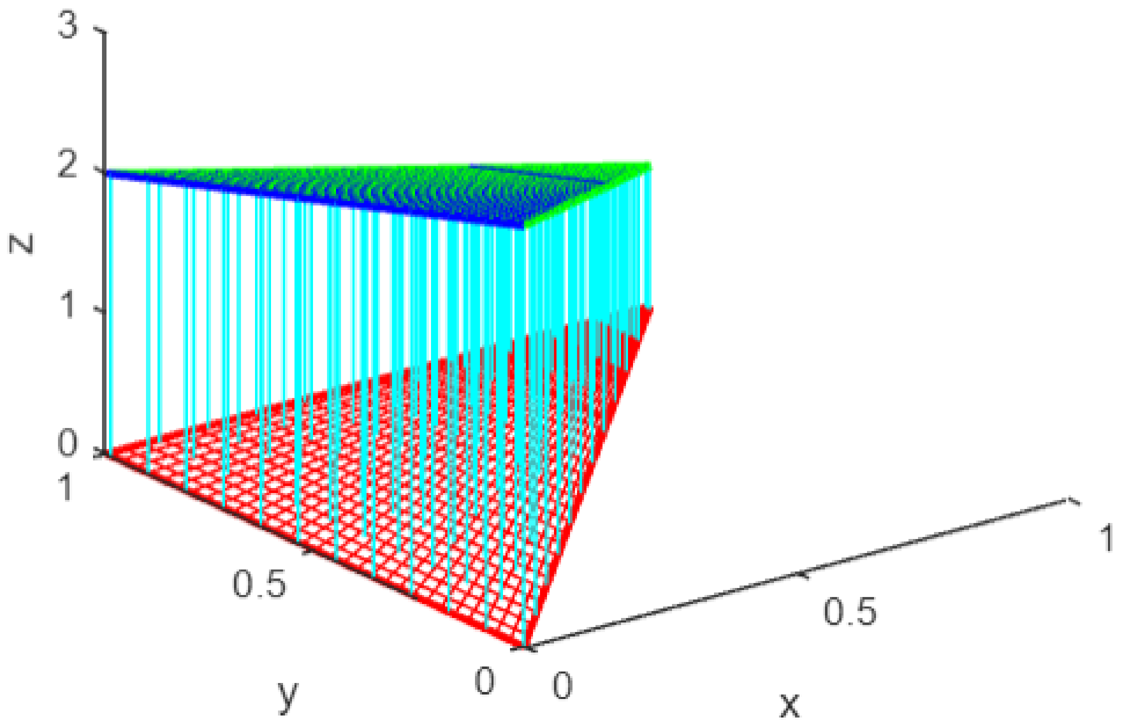
CODE-1:

INPUT:

```
syms x y z
int(int(3-x-y,x,y,1),y,0,1)
viewSolidone(z,0+0*x+0*y,3-x-y,x,y,1,y,0,1)
```

OUTPUT:

```
ans =

1
```



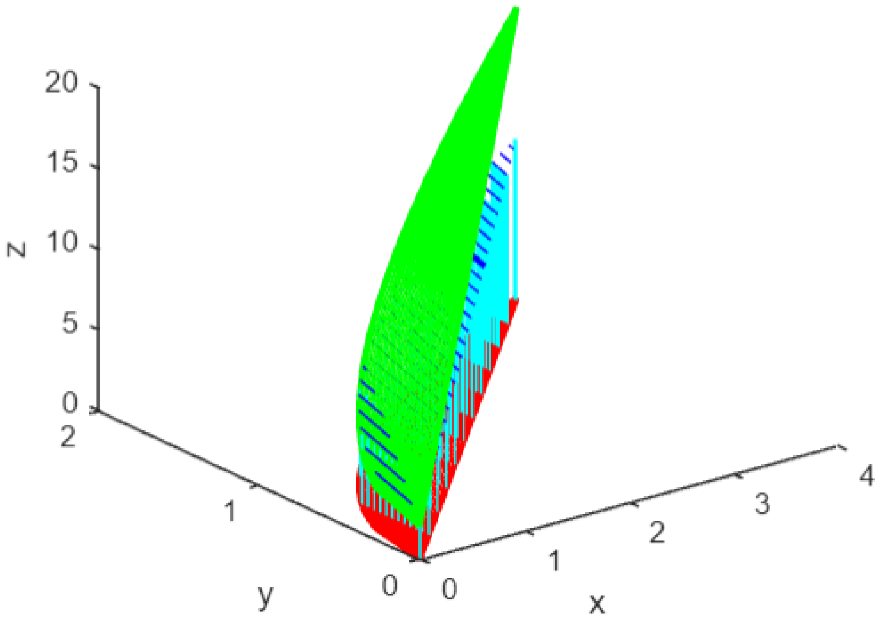
CODE-2:

INPUT:

```
clc
clear all
syms x y z
int(int(4*x+2,x,y/2,sqrt(y)),y,0,4)
viewSolidone(z,0+0*x+0*y,4*x+2,x,y/2,sqrt(y),y,0,4)
```

OUTPUT:

```
ans =

8
```



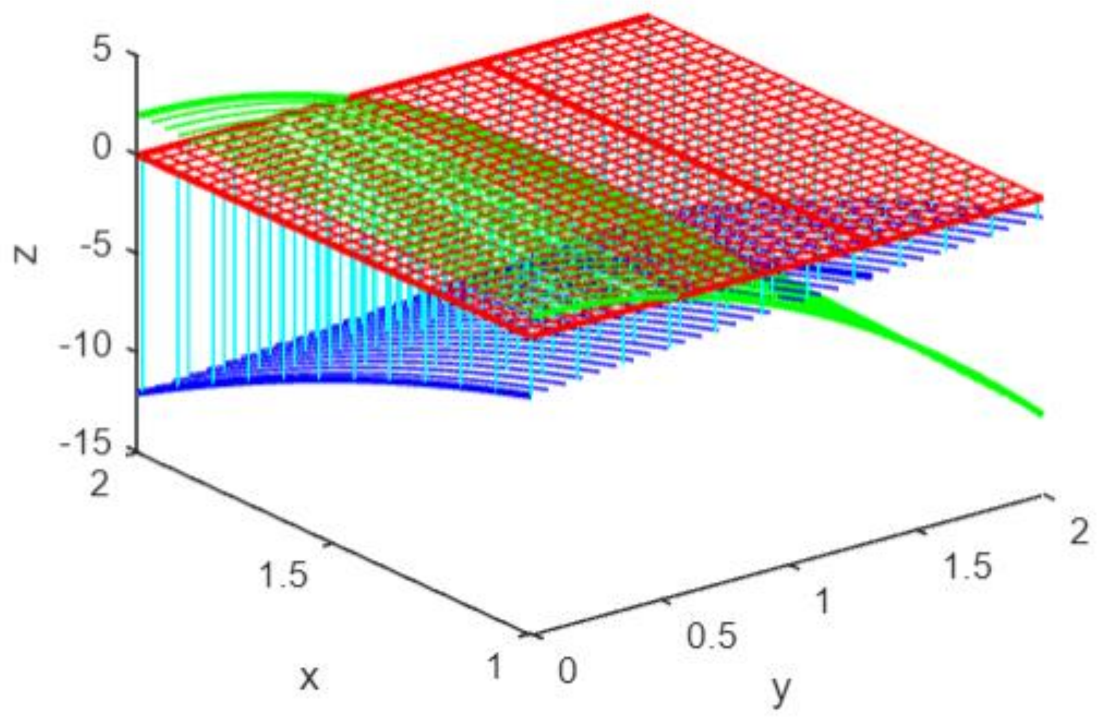
CODE-3:

INPUT:

```
clc
clear all
syms x y z
int(int(x-3*y^2,y,1,2),x,0,2)
viewSolidone(z,0+0*x+0*y,x-3*y^2,y,1+0*x,2+0*x,x,0,2)
```

OUTPUT:

```
ans =

-12
```



CODE 4:

INPUT:

```
clc
clear all
syms x y z
int(int(x.^2+y.^2,x,y/2,sqrt(y)),y,0,4)
```

OUTPUT:

```
ans =

216/35
```

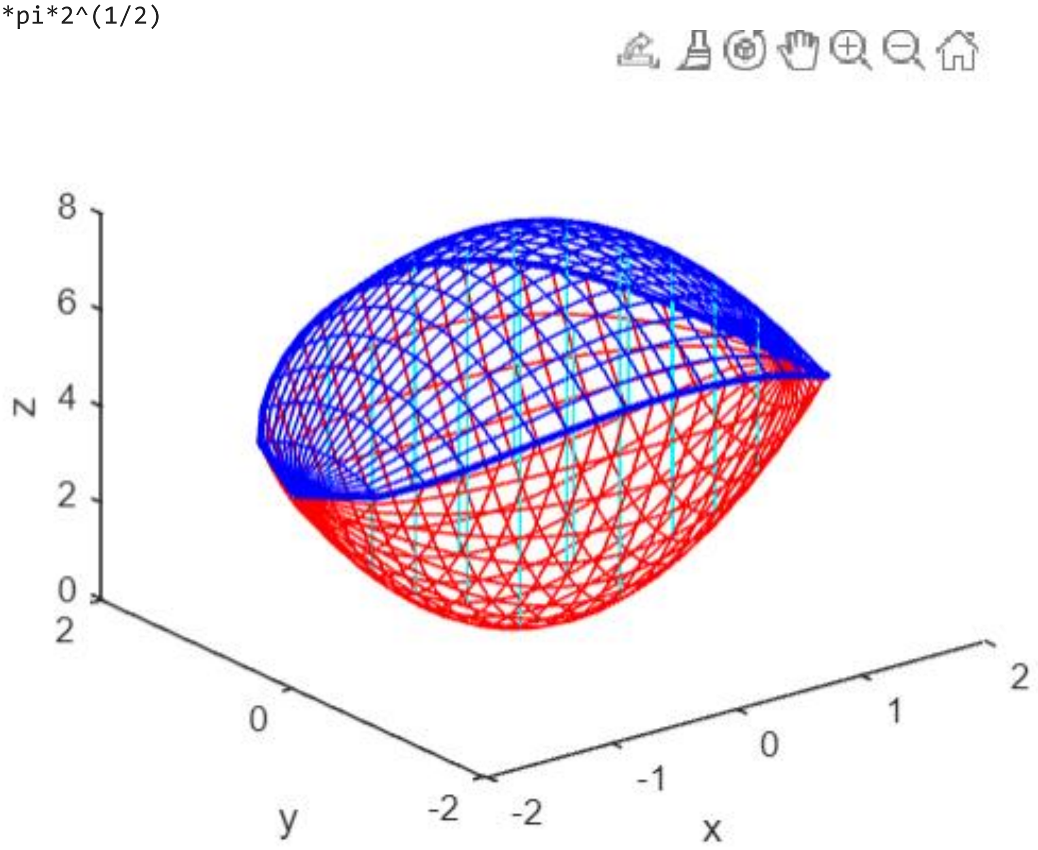
CODE-5:

INPUT:

```
clc
clear all
syms x y z
xa = -2;
xb = 2;
ya = -sqrt(2-x^2/2);
yb = sqrt(2-x^2/2);
za = x^2+3*y^2;
zb = 8-x^2-y^2;
int(int(int(1+0*z,z,za,zb),y,ya,yb),x,xa,xb)
viewSolid(z,za,zb,y,ya,yb,x,xa,xb)
```

OUTPUT:

```
ans =

8*pi*2^(1/2)
```



CODE-6:

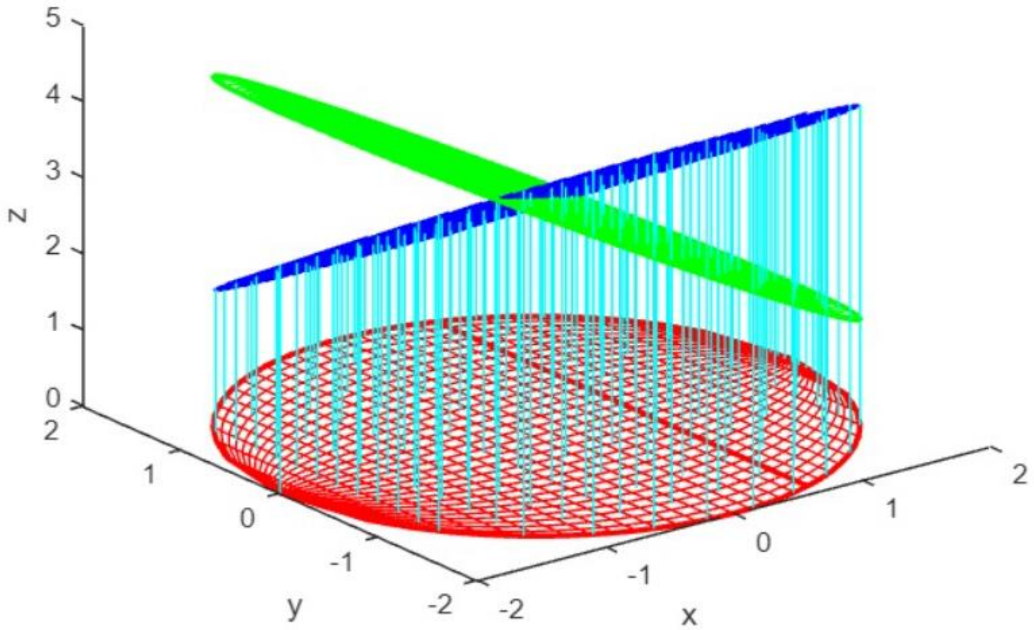
INPUT:

```
clc
```

```
clear all
syms x y z
xa = -sqrt(4-y^2);
xb = sqrt(4-y^2);
ya = -2;
yb = 2;
za = 0+0*x+0*y;
zb = 3-x-0*y;
int(int(int(1+0*z,z,za,zb),x,xa,xb),y,ya,yb)
viewSolid(z,za,zb,x,xa,xb,y,ya,yb)
```

OUTPUT:

```
ans =

12*pi
```



CODE-7:

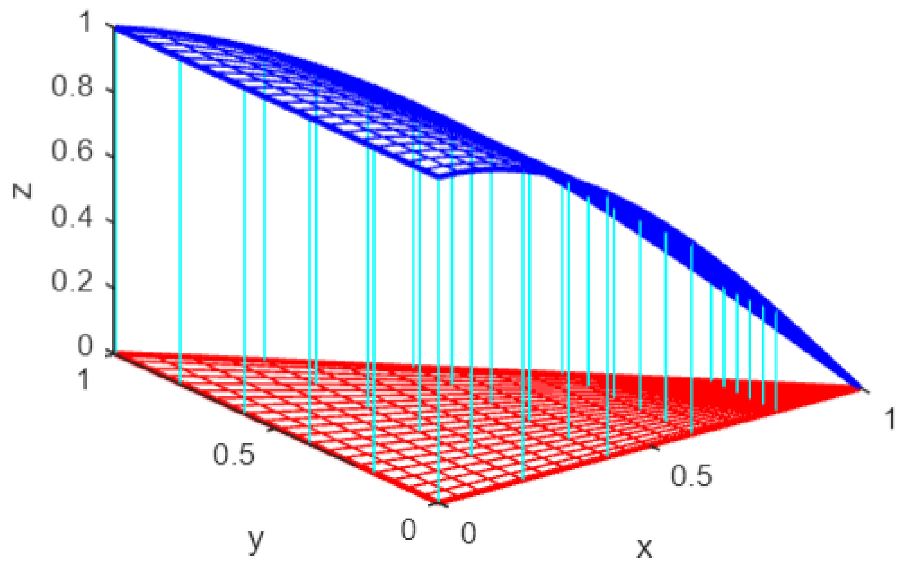
INPUT:

```
clc
clear all
syms x y z
```

```
xa=0;
xb=1;
ya=0+0*x;
yb=1-x;
za=0+0*x+0*y;
zb=cos(pi*x/2);
i=int(int(int(1+0*z,z,za,zb),y,ya,yb),x,xa,xb)
viewSolid(z,za,zb,y,ya,yb,x,xa,xb)
```

OUTPUT:

```
i =

4/pi^2
```



LAB-6-VECTOR CALCULUS IN MATLAB:

Syntax- inline,divergence,gradient,curl, Fcontour,ezcontour

CODE-1:

INPUT:

```
clc
clear all
fu=inline('x.^2+3*x+2','x');
fu(6)
```

OUTPUT:

```
ans =

    56
```

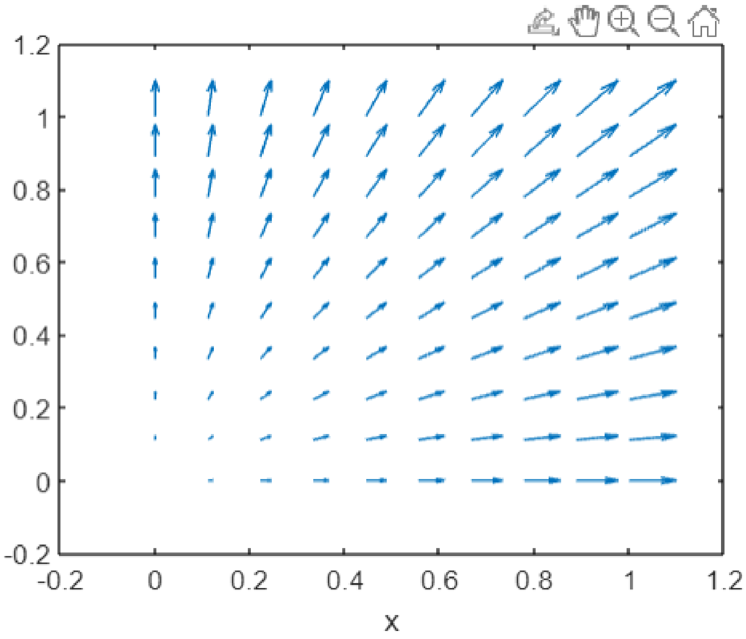
CODE-2: DRAW THE VECTOR FIELD FOR THE VECTOR

INPUT:

```
clear all
syms x y
f=input('enter the vector as i,j order in vector form:');
P=inline(vectorize(f(1)),'x','y');
Q=inline(vectorize(f(2)),'x','y');
x=linspace(0,1,10);
y=x;
[X,Y]=meshgrid(x,y);
U=P(X,Y);
V=Q(X,Y);
quiver(X,Y,U,V,1)
axis on
xlabel('x')
```

OUTPUT:

```
enter the vector as i,j order in vector form:
[x y]
```



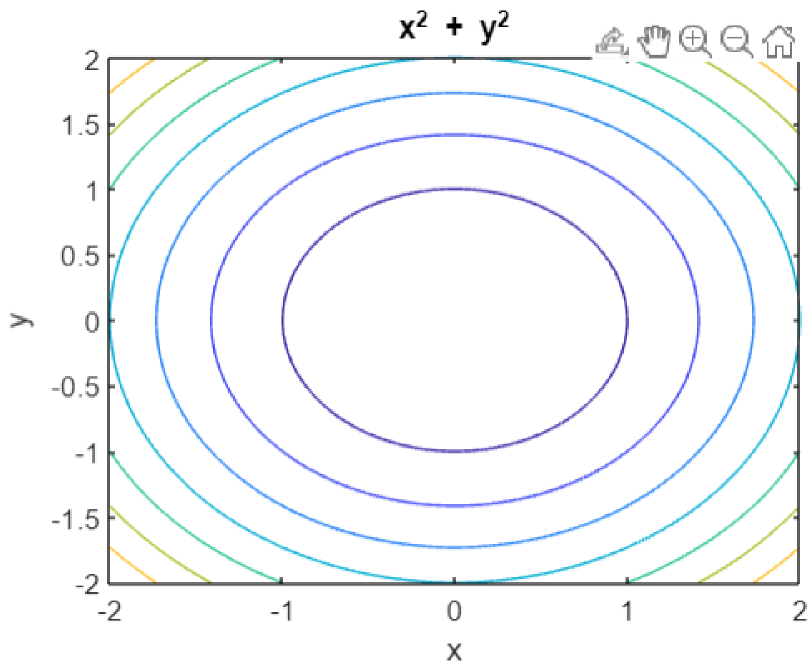
CODE-3: FINDING GRADIENT

INPUT:

```
clc
clear all
syms x y
f = input('enter the function f(x,y):')
f1 = diff(f,x);
f2 = diff(f,y);
P = inline(vectorize(f1),'x','y');
Q = inline(vectorize(f2),'x','y');
x = linspace(-2,2,10);
y = x;
[X,Y] = meshgrid(x,y);
U = P(X,Y);
V = Q(X,Y);
quiver(X,Y,U,V,1)
axis on
xlabel('x')
ylabel('y')
ezcontour(f,[-2,2])
```

OUTPUT:

```
enter the function f(x,y):
x^2+y^2

f =

x^2 + y^2
```



CODE-4:

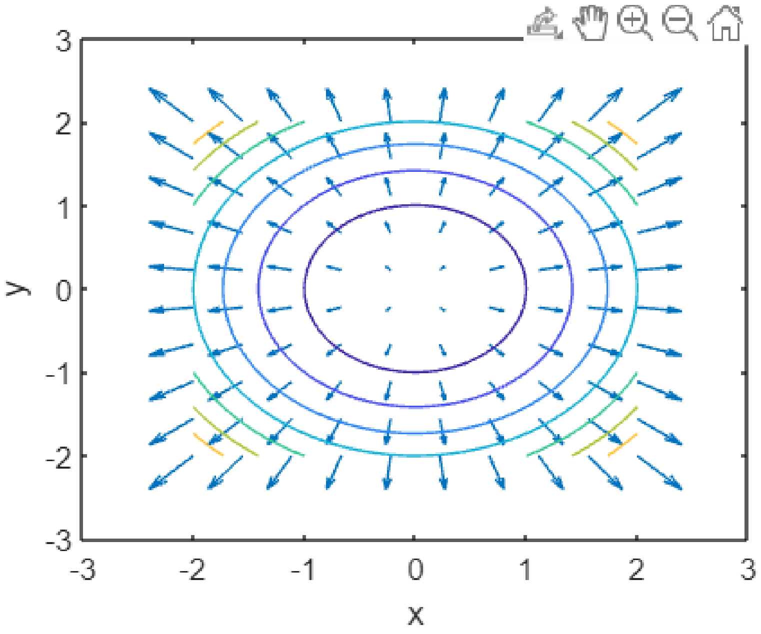
INPUT:

```
clear all
syms x y
f=input('enter the function f(x,y):')
grad=gradient(f,[x,y])
P(x,y)=grad(1);
Q(x,y)=grad(2);
x=linspace(-2,2,10);
y=x;
[X,Y]=meshgrid(x,y);
U=P(X,Y);
V=Q(X,Y);
quiver(X,Y,U,V,1)
axis on
xlabel('x')
ylabel('y')
hold on
fcontour(f,[-2,2])
```

OUTPUT:

```
enter the function f(x,y):
x^2+y^2

f =

x^2 + y^2
```

```
grad =

2*x
2*y
```



CODE-5:

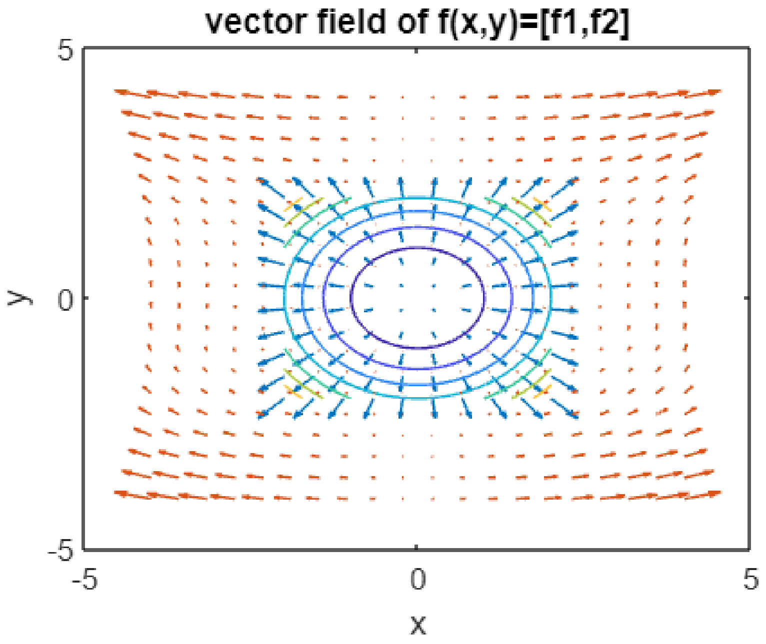
INPUT:

```
clc
clear all
syms x y
f=input('enter the 2D vector function in the form of [f1,f2]:')
div(x,y)=divergence(f,[x,y])
P(x,y)=f(1);
Q(x,y)=f(2);
x=linspace(-4,4,20);
y=x;
[X,Y]=meshgrid(x,y);
U=P(X,Y);
V=Q(X,Y);
shading interp
hold on
quiver(X,Y,U,V,1)
axis on
hold off
title('vector field of f(x,y)=[f1,f2]');
```

OUTPUT:

```
enter the 2D vector function in the form of [f1,f2]:
```

```
[x*y^2 x^2]

f =

[x*y^2, x^2]


div(x, y) =

y^2
```



CODE-6:

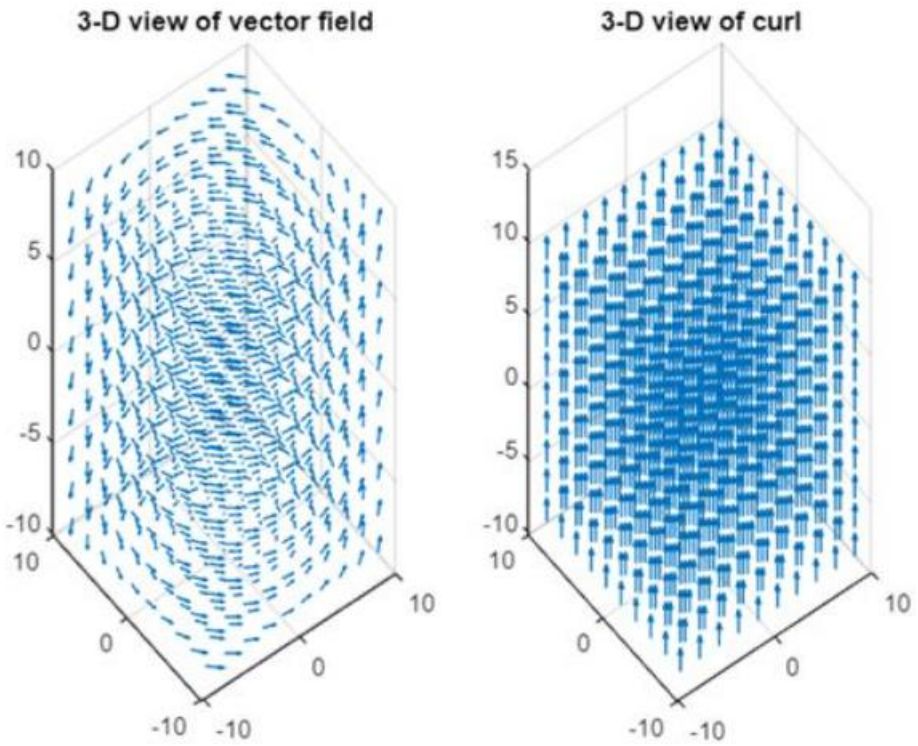
INPUT:

```
clc
clear all
syms x y z
f=input('enter the 3D vector function in the form of [f1,f2,f3]:')
P(x,y,z)=f(1);
Q(x,y,z)=f(2);
R(x,y,z)=f(3);
crl=curl(f,[x,y,z])
c1(x,y,z)=crl(1);
c2(x,y,z)=crl(2);
c3(x,y,z)=crl(3);
x=linspace(-9,9,10);
y=x;
z=x;
[X,Y,Z]=meshgrid(x,y,z);
U=P(X,Y,Z);
V=Q(X,Y,Z);
W=R(X,Y,Z);
CR1=c1(X,Y,Z);
CR2=c2(X,Y,Z);
```

```
CR3=c3(X,Y,Z);
figure;
subplot(1,2,1);
quiver(X,Y,Z,U,V,W);
title('3-D view of vector field');
subplot(1,2,2);
quiver3(X,Y,R,CR1,CR2,CR3);
title('3-D view of curl');
```

OUTPUT:

```
enter the 3D vector function in the form of [f1,f2,f3]:
[-y x 0]

f =

[-y, x, 0]


crl =

0
0
2
```



LAB-7 LINE INTEGRAL,WORK DONE AND SCALAR POTENTIAL

CODE-1:

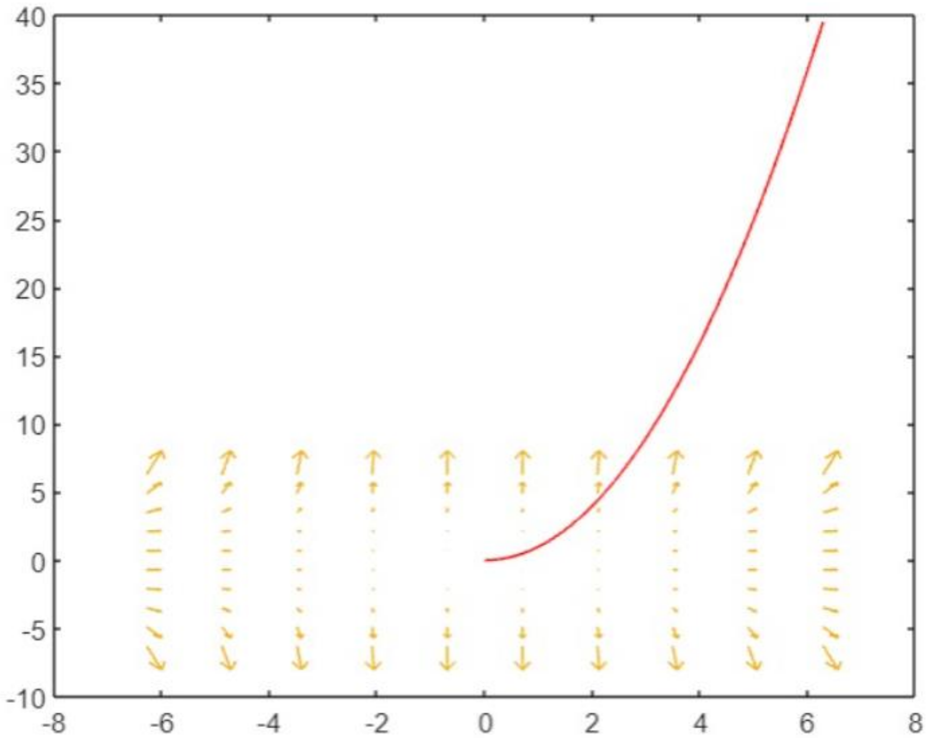
INPUT:

```
clc
clear all
syms x y t
f=input('Enter the componenets of 2D vector function [u,v] ')
r=input('Enter the x,y in parametric form')
I=input('Enter the limits of integration for t in the form [a,b]')
a=I(1);
b=I(2);
dr=diff(r,t);
F=subs(f,{x,y},r);
Fdr=sum(F.*dr);
I=int(Fdr,a,b);
P(x,y)=f(1);
Q(x,y)=f(2);
x1=linspace(-2*pi,2*pi,10);
y1=x1;
[X,Y]=meshgrid(x1,y1);
U=P(X,Y);
V=Q(X,Y);
quiver(X,Y,U,V,1)
hold on
t1=linspace(0,2*pi);
t2=linspace(0,2*pi);
x=subs(r(1),t1);
y=subs(r(2),t2);
plot(x,y,'r')
```

OUTPUT:

```
Enter the componenets of 2D vector function [u,v]
[x^2 y^3]

f =

[x^2, y^3]

Enter the x,y in parametric form
[t t^2]

r =

[t, t^2]

Enter the limits of integration for t in the form [a,b]
[0,1]
```

```
I =

     0     1
```



CODE-2:

INPUT:

```
clc
clear all
syms x y z real
f=input('enter the vector:');
curl_f=curl(f,[x,y,z]);
if(curl_f==[0 0 0])
    f=potential(f,[x,y,z])
else
    sprintf('curl_f is not equal to 0')
end
```

OUTPUT:

```
enter the vector:
[x^2 y^2 z^2]

f =

x^3/3 + y^3/3 + z^3/3
```